We Claim:

1. A method of designing a circuit having at least one hierarchical block which requires block specific test patterns to facilitate quiescent current testing of the circuit, comprising, for each block requiring block specific test patterns:

configuring the block and any embedded blocks located one level down in design hierarchy in quiescent current test mode in which input peripheral memory elements are configured in internal test mode and output peripheral memory elements are configured in external test mode;

generating quiescent current test patterns which do not result in elevated quiescent current levels and which include a bit for all memory elements in said block and any peripheral memory elements in any embedded blocks located one level down in design hierarchy; and

if said block contains embedded blocks, synchronizing each test pattern with a corresponding test pattern generated for embedded blocks so that test patterns loaded in scan chains in the block are consistent with test patterns loaded in scan chains in said embedded blocks.

2. A method as defined in claim **1**, further including merging block level test patterns using a circuit consolidated quiescent current scan chain description to provide a consolidated set of circuit quiescent current test patterns for use in quiescent current testing of said circuit.

3. A method as defined in claim **2**, said merging block level test patterns further including, for corresponding patterns of block test patterns, assigning a value to each memory element on each scan chain of said consolidated scan chain description, said assigning including extracting the value from the pattern of the block containing the memory element when the memory element requires controllability for internal testing only and extracting the value from the pattern of the parent block of the block containing the memory element when the memory element requires controllability for both internal and external testing.

4. A method as defined in claim **3**, said merging including, for memory elements which share a source with memory elements in a parent or embedded block, selecting values for said memory elements from the pattern generated for the parent block.

5. A method as defined in claim **4**, further including modifying the test pattern of said embedded block to correspond to values selected from the pattern of the parent block.

6. A method as defined in claim **1**, said synchronizing including assigning to peripheral memory elements of embedded blocks, specific bits assigned to the peripheral memory elements in a corresponding test pattern generated for said embedded blocks.

7. A method as defined in claim **1**, said synchronizing including determining whether a memory element segment in the block shares a common input with a segment in an embedded block and, if so, assigning to memory elements in the segment, and in any dependent segments, any specific bits assigned to memory elements in the embedded block segment, and any dependent segments, for a corresponding test pattern.

8. A method as defined in claim **7**, said determining whether a segment in the block shares a common input including determining whether the distance in bits from the block serial input of said block segment is the same as the distance in bits from the embedded block serial input to its block serial input.

9. A method as defined in claim **1**, said synchronization including modifying a test pattern previously generated for an embedded block to incorporate therein a specific bit of a block being processed.

10. A method as defined in claim **1**, said generating quiescent current patterns for blocks in said circuit beginning with blocks at a lowest level of hierarchy and proceeding in sequence through each level of design hierarchy to a highest level of hierarchy containing a top-level block.

11. A method as defined in claim **1**, further including, prior to generating said test patterns, performing a static analysis of said block and of a simplified model of embedded block to determine whether no circuit states would cause elevated quiescent current levels.

12. A method as defined in claim **11**, further including, if said static analysis identified circuit states which cause an elevated quiescent current level, then, after generating a test pattern for a block, examining the test pattern to determine whether it would result in an elevated quiescent current level and discarding the test pattern if an elevated current level would result and storing the test pattern if no elevated current level would result.

13. A method as defined in claim **1**, further including calculating a fault coverage for said block.

14. A method as defined in claim **13**, said calculating a fault coverage including calculating the total fault coverage for said block by calculating the weighted sum of the fault coverages of said block and of embedded blocks located one level of design hierarchy.

15. A method as defined in claim **14**, said weighted sum being based on the number of gates in each of said block and embedded blocks located one level down in design hierarchy.

16. A method as defined in claim **15**, further including determining and storing the number of gates of the block and the cumulative fault coverage for each test pattern.

17. A method as defined in claim **16**, said calculating fault coverage including using a pseudo stuck-at fault coverage model or a toggle fault coverage model.

18. A method as defined in claim **1**, further including generating and storing a simplified model of said block.

19. A method as defined in claim **1**, further including simulating said test patterns to verify their correctness.

20. A method as defined in claim **1**, wherein, prior to generating quiescent current test patterns for said block, arranging scannable memory elements in each block into quiescent current scan chain segments in which the segments are arranged on a scan path which originates from or includes a block serial input by connecting a segment serial input to: a block input, the serial output of another segment or the serial output of a segment located in an embedded block one level down in design hierarchy and by connecting block serial inputs of embedded blocks to block serial inputs.

21. A method as defined in claim **20**, further including arranging internal memory element segments of a block on a scan path which originates from a block input by connecting the serial input of internal segments to a block input, the serial output of another segment in the block or the serial output of a peripheral segment located in an embedded block and connecting the serial input of scan paths which include internal segments of embedded blocks to a block serial input.

22. A method as defined in claim **21**, further including, arranging peripheral memory element segments of a block on a scan path which does not include any branches and which originates from or includes a block serial input by connecting the serial input of each peripheral segment to a block input, or to the serial output of another peripheral segment and connecting the serial input of scan paths consisting exclusively of peripheral segments of embedded blocks are connected to to a block serial input, to the serial output of another segment within the block or to the serial output of a peripheral segment located in an embedded block.

23. A method as defined in claim **20**, further including, arranging peripheral memory element segments of a block on a scan path which does not include any branches and which originates from or includes a block serial input by connecting the serial input of each peripheral segment to a block input or to the serial output of another peripheral segment and connecting the serial input of scan paths consisting exclusively of peripheral segments of embedded blocks to a block serial input, to the serial output of another segment within the block or to the serial output of a peripheral segment located in an embedded block.

24. A method as defined in claim **23**, further including generating and storing on a computer readable storage medium a scan chain description for each said scan path in the block.

25. A method as defined in claim **24**, further including consolidating said scan chain descriptions into a consolidated scan chain description for use in performing quiescent current testing of said circuit.

26. A method as defined in claim **1**, further including storing said quiescent current test patterns for each block on a computer readable storage medium.

27. A method of designing a circuit having at least one hierarchical block which requires block specific test patterns to facilitate quiescent current testing of the circuit, comprising:

arranging scannable memory elements in each block requiring block specific test patterns into quiescent current scan chain segments in which the segments are arranged on a scan path which originates from or includes a block serial input by connecting a segment serial input to a block serial input, to the serial output of another segment in the block or to the serial output of a segment located in an embedded block one level down in design hierarchy and by connecting the block serial input of embedded blocks to the block serial input;

configuring the block and any embedded blocks located one level down in design hierarchy in quiescent current test mode in which input peripheral memory elements are configured in internal test mode and output peripheral memory elements are configured in external test mode;

generating quiescent current test patterns which do not result in elevated quiescent current levels, each test pattern including a bit for all memory elements in the block and peripheral memory elements in any embedded blocks located one level down in design hierarchy; and

synchronizing each test pattern with a corresponding test pattern generated for embedded blocks so that test patterns loaded in scan chains in the block are consistent with test patterns loaded in scan chains in said embedded blocks,

said synchronizing including assigning to peripheral memory elements of embedded blocks, specific bits assigned to the peripheral memory elements in a corresponding test pattern generated for said embedded blocks and determining whether a memory element segment in the block shares a common input with a segment in an embedded block and, if so, assigning to memory elements in the segment, and in any dependent segments, any specific bits assigned to memory elements in the embedded block segment and any dependent segments for a corresponding test pattern.

28. A method as defined in claim **27**, further including, after test patterns have been generated for all selected blocks, merging block level test patterns using a circuit consolidated quiescent current scan chain description to provide a consolidated set of circuit quiescent current test patterns for use in quiescent current testing of said circuit.

29. A method as defined in claim **28**, said merging block level test patterns further including, for corresponding patterns of block test patterns, assigning a value to each memory element on each scan chain of said consolidated scan chain description, said assigning including extracting the value from the pattern of the block containing the memory element when the memory element requires controllability for internal testing only and extracting the value from the pattern of the parent block of the block containing the memory element when the memory element requires controllability for both internal and external testing.

30. A method as defined in claim **29**, said merging including, for memory elements which share a source with memory elements in a parent or embedded block, selecting values for said memory elements from the pattern generated for the parent block.

31. A method as defined in claim **30**, further including modifying the test pattern of said embedded block to correspond to values selected from the pattern of the parent block.

32. A method as defined in claim **29**, further including, after generating a test pattern, examining the test pattern to determine whether it would result in an elevated quiescent current level, discarding the test pattern if an elevated current level would result and storing the test pattern if no elevated current level would result.

33. A method as defined in claim **32**, further including arranging internal memory element segments of a block on a scan path which originates from a block input by connecting the serial input of internal segments to a block input, to the serial output of another segment in the block or to the serial output of a peripheral segment located in an embedded block without creating branches on the scan path and connecting the serial input of scan paths which include internal segments of embedded blocks to a block serial input.

34. A method as defined in claim **33**, further including, arranging peripheral memory element segments of a block on a scan path which originates from or includes a block serial input by connecting the serial input of each peripheral segment to a block input, to or to the serial output of another peripheral segment and connecting the serial input of scan paths consisting exclusively of peripheral segments of embedded blocks are connected to to a block serial input, to the serial output of another segment (internal or peripheral) within the block or to the serial output of a peripheral segment located in an embedded block without any branches on the scan path.

35. A method as defined in claim **34**, further including, prior to generating said test patterns, analyzing said block to determine circuit states which would not cause elevated quiescent current levels, said generating quiescent current test patterns using bit patterns corresponding to said circuit states.

36. A method as defined in claim **35**, said analyzing said block including performing a static analysis of said block to determine whether no circuit states cause elevated current values.

37. A method as defined in claim **35**, said generating test patterns including using a predetermined simplified model for selected embedded blocks.

38. A method as defined in claim **37**, further including calculating a fault coverage for said block.

39. A method as defined in claim **38**, said calculating a fault coverage comprising calculating a weighted fault coverage.

40. A method as defined in claim **39**, further including generating and storing a simplified model of said block.

41. A method as defined in claim **40**, further including simulating said test patterns to verify their correctness.

42. A program product of designing a circuit having at least one hierarchical block which requires block specific test patterns to facilitate quiescent current testing of the circuit, comprising:

a computer readable storage medium;

means recorded on said storage medium for, for each block requiring block specific test patterns, configuring the block and any embedded blocks located one level down in design hierarchy in quiescent current test mode in which input peripheral memory elements are configured in internal test mode and output peripheral memory elements are configured in external test mode;

means recorded on said storage medium for generating quiescent current test patterns which do not result in elevated quiescent current levels and which include a bit for all memory elements in said block and peripheral memory elements in embedded blocks located one level down in design hierarchy; and

means recorded on said storage medium for synchronizing each test pattern with a corresponding test pattern generated for embedded blocks so that test patterns loaded in scan chains in the block are consistent with test patterns loaded in scan chains in said embedded blocks.

43. A program product as defined in claim **42**, said means for synchronizing being operable to assign to peripheral memory elements of embedded blocks, specific bits assigned to the peripheral memory elements in a corresponding test pattern generated for said embedded blocks.

44. A program product as defined in claim **42**, said means for synchronizing including means for determining whether a memory element segment in the block shares a common input with a segment in an embedded block and, if so, assign to memory elements in the segment, and in any dependent segments, any specific bits assigned to memory elements in the embedded block segment, and any dependent segments, for a corresponding test pattern.

45. A program product as defined in claim **44**, said means for determining including means for determining whether the distance in bits from the block serial input of said block segment is the same as the distance in bits from the embedded block serial input to its block serial input.

46. A program product as defined in claim **42**, said means for synchronization including means for modifying a test pattern previously generated for an embedded block to incorporate therein a specific bit of a block being processed.

47. A program product as defined in claim **42**, said means for generating quiescent current patterns for blocks in said circuit beginning with blocks at a lowest level of hierarchy and proceeding in sequence through each level of design hierarchy to a highest level of hierarchy containing a top-level block.

48. A program product as defined in claim **42**, further including means recorded on said medium for performing a static analysis of said block and of a simplified model of embedded blocks one level down in design hierarchy to determine whether no circuit states would cause elevated quiescent current levels.

49. A program product as defined in claim **48**, said means for generating further including means recorded on said medium for examining a test pattern to determine whether the test pattern would result in an elevated quiescent current level, discarding the test pattern if an elevated current level would result and storing the test pattern if no elevated current level would result.

50. A program product as defined in claim **42**, further including means recorded on said storage medium for calculating a fault coverage for said block.

51. A program product as defined in claim **50**, said means for calculating a fault coverage including calculating a total fault coverage for the block by calculating the weighted sum of the fault coverages of said block and of embedded blocks located one level of design hierarchy.

52. A program product as defined in claim **51**, said means for calculating being operable to calculate a weighted sum based on the number of gates in each of said block and embedded blocks located one level down in design hierarchy.

53. A program product as defined in claim **52**, further including means recorded on said storage medium for determining and storing the number of gates of the block and the cumulative fault coverage for each test pattern.

54. A program product as defined in claim **53**, said means for calculating a fault coverage being operable to use a pseudo stuck-at fault coverage model or a toggle fault coverage model.

55. A program product as defined in claim **42**, further including means recorded on said storage medium for generating and storing a simplified model of said block.

56. A program product as defined in claim **42**, further including means recorded on said storage medium for simulating said test patterns to verify their correctness.

57. A program product as defined in claim **42**, further including means recorded on said storage medium for arranging scannable memory elements in said block into quiescent current scan chain segments in which the segments are arranged on a scan path which originates from or includes a block serial input by connecting a segment serial input to: a block input, the serial output of another segment or the serial output of a segment located in an embedded block one level down in design hierarchy and by connecting block serial inputs of embedded blocks to block serial inputs.

58. A program product as defined in claim **57**, said means for arranging being operable to arrange internal memory element segments of a block on a scan path which originates from a block input by connecting the serial input of internal segments to a block input, the serial output of another segment in the block or the serial output of a peripheral segment located in an embedded block and connecting the serial input of scan paths which include internal segments of embedded blocks to a block serial input.

59. A program product as defined in claim **58**, said means for arranging being operable to arrange peripheral memory element segments of a block on a scan path which does not include any branches and which originates from or includes a block serial input by connecting the serial input of each peripheral segment to a block input, or to the serial output of another peripheral segment and connecting the serial input of scan paths consisting exclusively of peripheral segments of embedded blocks are connected to to a block serial input, to the serial output of another segment within the block or to the serial output of a peripheral segment located in an embedded block.

60. A program product as defined in claim **58**, said means for arranging being operable to arrange peripheral memory element segments of a block on a scan path which does not include any branches and which originates from or includes a block serial input by connecting the serial input of each peripheral segment to a block input or to the serial output of another peripheral segment and connecting the serial input of scan paths consisting exclusively of peripheral segments of embedded blocks to a block serial input, to the serial output of another segment within the block or to the serial output of a peripheral segment located in an embedded block.

61. A program product as defined in claim **60**, further including means recorded on said storage medium for generating and storing on a computer readable storage medium a scan chain description for each said scan path in the block.

62. A program product as defined in claim **61**, further including means recorded on said storage medium for consolidating said scan chain descriptions into a consolidated scan chain description for use in performing quiescent current testing of said circuit.

63. A program product as defined in claim **42**, further including means recorded on said storage medium for storing said quiescent current test patterns for each block on a computer readable storage medium.

64. A program product as defined in claim **42**, further including means recorded on said storage medium for merging block level test patterns with a consolidated scan chain description for the circuit to provide a consolidated set of circuit quiescent current test patterns for use in quiescent current testing of said circuit.

65. A program product as defined in claim **64**, said means for merging block level test patterns further including, for corresponding patterns of block test patterns, means for assigning a value to each memory element on each scan chain of said consolidated scan chain description, said means for assigning extracting the value from the pattern of the block containing the memory element when the memory element requires controllability for internal testing only and extracting the value from the pattern of the parent block of the block containing the memory element when the memory element requires controllability for both internal and external testing.

66. A method as defined in claim **65**, said merging including, for memory elements which share a source with memory elements in a parent or embedded block, selecting values for said memory elements from the pattern generated for the parent block.

67. A method as defined in claim **66**, further including modifying the test pattern of said embedded block to correspond to values selected from the pattern of the parent block.

68. A program product of designing a circuit having at least one hierarchical block which requires block specific test patterns to facilitate quiescent current testing of the circuit, comprising:

a computer readable storage medium;

means recorded on said storage medium for arranging scannable memory elements in each block requiring block specific test patterns into quiescent current scan chain segments in which the segments are arranged on a scan path which originates from or includes a block serial input by connecting a segment serial input to a block serial input, to the serial output of another segment in the block or to the serial output of a segment located in an embedded block one level down in design hierarchy and by connecting the block serial input of embedded blocks to the block serial input;

means recorded on said storage medium for configuring the block and any embedded blocks located one level down in design hierarchy in quiescent current test mode in which input peripheral memory elements are configured in internal test mode and output peripheral memory elements are configured in external test mode;

means recorded on said storage medium for generating quiescent current test patterns which do not result in elevated quiescent current levels, each test pattern including a bit for all memory elements in the block and peripheral memory elements in any embedded blocks located one level down in design hierarchy; and

means recorded on said storage medium for synchronizing each test pattern with a corresponding test pattern generated for embedded blocks so that test patterns loaded in scan chains in the block are consistent with test patterns loaded in scan chains in said embedded blocks,

said means for synchronizing including means for assigning to peripheral memory elements of embedded blocks, specific bits assigned to the peripheral memory elements in a corresponding test pattern generated for said embedded blocks and for determining whether a memory element segment in the block shares a common input with a segment in an embedded block and, if so, for assigning to memory elements in the segment, and in any dependent segments, any specific bits assigned to memory elements in the embedded block segment and any dependent segments for a corresponding test pattern.

69. A program product as defined in claim **68**, further including means recorded on said storage medium for merging block level test patterns with a consolidated scan chain description for the circuit to provide a consolidated set of circuit quiescent current test patterns for use in quiescent current testing of said circuit, said means for merging block level test patterns further including, for corresponding patterns of block test patterns, means for assigning a value to each memory element on each scan chain of said consolidated scan chain description, said means for assigning extracting the value from the pattern of the block containing the memory element when the memory element requires controllability for internal testing only and extracting the value from the pattern of the parent block of the block containing the memory element when the memory element requires controllability for both internal and external testing.

70. A program product as defined in claim **68**, further including means recorded on said storage medium for, after generating a test pattern, examining the test pattern to determine whether it would result in an elevated quiescent current level, discarding the test pattern if an elevated current level would result and storing the test pattern if no elevated current level would result.

71. A program product as defined in claim **70**, said means for arranging being further operable to arrange internal memory element segments of a block on a scan path which originates from a block input by connecting the serial input of internal segments to a block input, to the serial output of another segment in the block or to the serial output of a peripheral segment located in an embedded block without creating branches on the scan path and connecting the serial input of scan paths which include internal segments of embedded blocks to a block serial input.

72. A program product as defined in claim **71**, said means for arranging being further operable to arrange peripheral memory element segments of a block on a scan path which originates from or includes a block serial input by connecting the serial input of each peripheral segment to a block input, to or to the serial output of another peripheral segment and connecting the serial input of scan paths consisting exclusively of peripheral segments of embedded blocks are connected to to a block serial input, to the serial output of another segment (internal or peripheral) within the block or to the serial output of a peripheral segment located in an embedded block without any branches on the scan path.

73. A program product as defined in claim **72**, further including means recorded on said storage medium for analyzing said block to determine circuit states which would not cause elevated quiescent current levels, said generating quiescent current test patterns using bit patterns corresponding to said circuit states.

74. A program product as defined in claim **73**, said means for analyzing said block including means for performing a static analysis of said block to determine whether no circuit states cause elevated current values.

75. A program product as defined in claim **73**, said means for generating test patterns using a simplified model for selected embedded blocks.

76. A program product as defined in claim **75**, further including means recorded on said storage medium for calculating a fault coverage for said block.

77. A program product as defined in claim **76**, said means for calculating a fault coverage being operable to calculate a weighted fault coverage.

78. A program product as defined in claim **77**, further including means recorded on said storage medium for generating and storing a simplified model of said block.

79. A program product as defined in claim **78**, further including means recorded on said storage medium for simulating said test patterns to verify their correctness.